



#15

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application: **Kult**

Serial No.: **09/096,939**

Filed: **June 12, 1998**

For: **A SYSTEM AND METHOD  
FOR RESOURCE MANAGEMENT**

§  
§  
§  
§  
§  
§  
§  
§  
§

Group Art Unit: **2743**

Examiner: **Barnie, R.**

Attorney Docket No.: **CDR-97-031**

**Certificate of Mailing Under 37 C.F.R. §1.8(a)**

I hereby that certify this correspondence is being deposited with the United States Postal Service as First Class mail in an envelope addressed to: Assistant Commissioner of Patents, Washington, D.C. 20231 on February 12, 2002.

By: Marsha S. Kappus  
Marsha S. Kappus

Assistant Commissioner for Patents  
Washington, D.C. 20231

ATTENTION: Board of Patent Appeals and Interferences

RECEIVED  
MAR 07 2002  
Technology Center 2600

**APPELLANT'S REPLY BRIEF (37 C.F.R. 1.193(b)(1))**

This brief is in furtherance of the Notice of Appeal, filed in this case on **January 12, 2001** and in response to Examiner's Answer filed on **December 18, 2001**.

The fees required under § 1.17(c), and any required petition for extension of time for filing this brief and fees therefor, are dealt with in the accompanying TRANSMITTAL OF APPEAL BRIEF.

This brief is transmitted in triplicate. (37 C.F.R. 1.192(a))

03/05/2002 SFELEKE1 00000040 09096939

01 FC:120

320.00 CP

This reply brief contains replies to Examiner's Answer under the following headings, and in the order set forth below (37 C.F.R. 1.192(c)):

- I REAL PARTY INTEREST
- II RELATED APPEALS AND INTERFERENCES
- III STATUS OF CLAIMS
- IV STATUS OF AMENDMENTS
- V SUMMARY OF INVENTION
- VI ISSUES
- VII GROUPING OF CLAIMS
- VIII ARGUMENTS
  - ARGUMENT: VIIIA REJECTIONS UNDER 35 U.S.C. 112
  - ARGUMENT: VIIIB REJECTIONS UNDER 35 U.S.C. 102
- IX APPENDIX OF CLAIMS INVOLVED IN THE APPEAL

**I. REAL PARTIES IN INTEREST (37 C.F.R. 1.192(c)(1))**

The real party in interest in this appeal is the following party: WorldCom, Inc.

## **II. RELATED APPEALS AND INTERFERENCES (37 C.F.R. 1.192(c)(2))**

With respect to other appeals or interferences that will directly affect, or be directly affected by, or have a bearing on the Board's decision in the pending appeal, there are no such appeals or interferences.

### **III. STATUS OF CLAIMS (37 C.F.R. 1.192(c)(3))**

#### **A. TOTAL NUMBER OF CLAIMS IN APPLICATION**

Claims in the application are: 20

#### **B. STATUS OF ALL THE CLAIMS IN APPLICATION**

1. Claims canceled: None
2. Claims withdrawn from consideration but not canceled: None
3. Claims pending: **1 - 20**
  1. Claims allowed: **5**
  1. Claims rejected: **1 - 4, and 6 - 20**

#### **C. CLAIMS ON APPEAL**

The claims on appeal are: **1 - 4, and 6 - 20**

#### **IV. STATUS OF AMENDMENTS (37 C.F.R. 1.192(c)(4))**

An after final amendment was filed but not entered by the Examiner.

## V. SUMMARY OF INVENTION (37 C.F.R. 1.192(c)(5))

The present invention is directed to managing resources in a telecommunication network that are related to or accessible by a switch, such as Intelligent Service Networks (ISNs) **102**

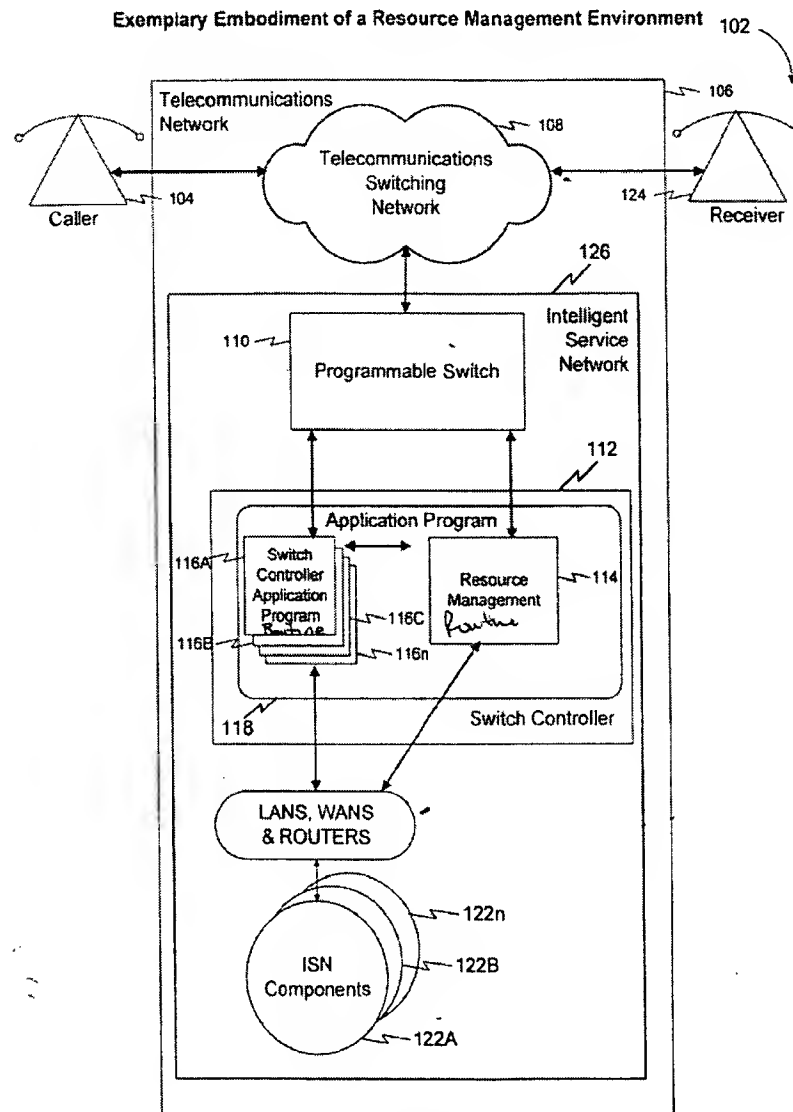


FIG. 1

shown in **FIG. 1**, reproduced herein for convenience.

Prior art telecommunications networks require that, for each unique resource, a requestor understands and utilizes particular Application Programmer Interfaces (APIs) in order for the requestor to access or update the information related to the resource. Typically, all resource information is stored and managed by the particular resource. Thus, each time a requestor

desires resource information or to update resource information for a particular resource, according to the prior art, the requestor must express a query using an API understood by the particular resource and then transmit the query to the resource itself for processing. These prior art APIs are resource-based.

By contrast, the present invention uses a resource management routine which contains individual resource manager correlating to the resources being managed by the resource management routine. Each resource manager comprises a table of data related to the resource to which it corresponds and to a generic resource management API for acquiring data from the table. All of the generic resource management APIs are standard resource APIs enabling the

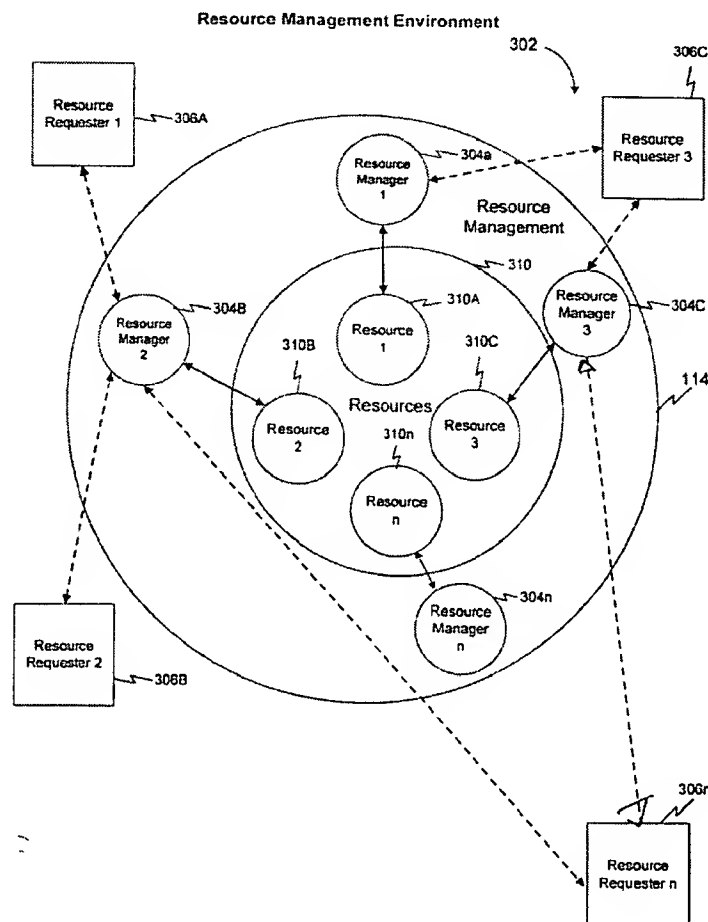


FIG. 3

requestor to query a resource without needing to know each APIs that is understood by each particular resource (see page 3, line 22 to 5, line 10).

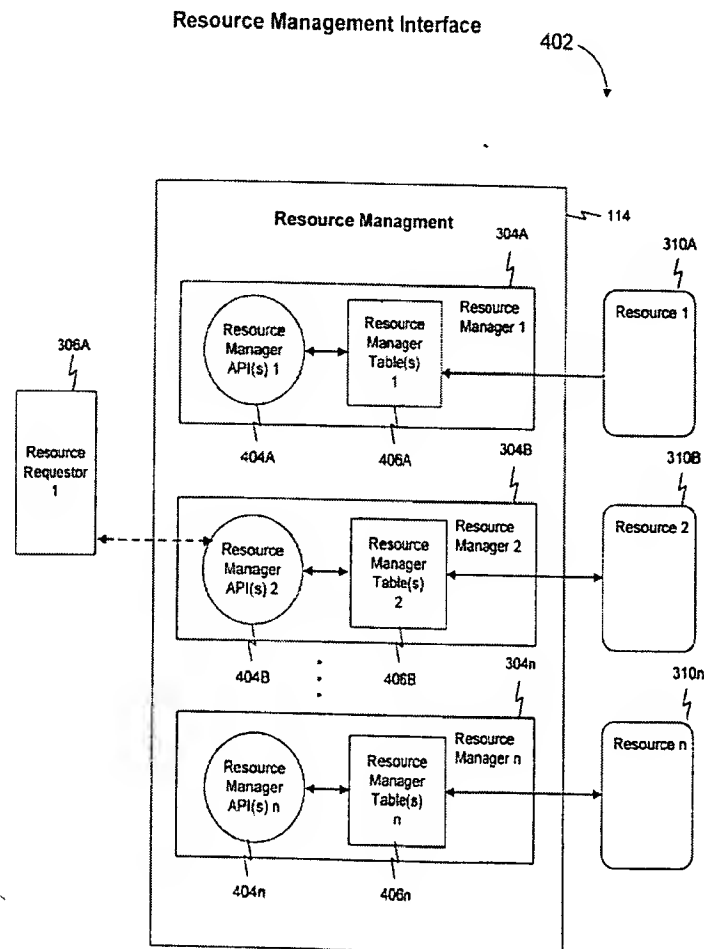
Resource management routine **114** managing all queries for resources associated with, for instance, an ISN. When used in ISN **116**, resource management routine **114** resides in the memory of switch controller **112**.

With regard to **FIG. 3**, resource management routine **114** uses individual resource managers **304A - 304n** to handle queries originating from any of requestors **306A - 306n** directed to respective one of corresponding resources **310A - 310n**. Resources **310A - 310n** may be of different type categories (*i.e.*, internal operational resources, external components, or applications processing data). With the inclusion of resource managers **304A - 304n** (within resource management routine **114**), requestors **306A - 306** do not interface directly with resources **310A - 310n**. Instead, each of resource managers **304A - 304n** contain the resource information and procedures necessary for interfacing and obtaining information from the respective resources **310A - 310n**. In this case, requestors **306A - 306** actually interface with a particular resource manager **304** associated with a particular resource **310** that requester **306** intends to interface with. Thus, it can be seen that resource management routine **114** serves as a protective layer between requestors **306A - 306n** and resources **310A - 310n** as is more clearly illustrated in **FIG. 3**. However, in practice, an individual resource manager actually interfaces between its corresponding resource and any of requestor needing to interface with the resource, as also can be appreciated from **FIG. 3** but is more apparent from **FIG. 4**.

According to the present invention, each of resource managers **304A - 304n** serves two roles. They store information about a particular corresponding resource **310** (in resource manager table **406**), and provide a standard mechanism for all requestors **306** to interface with and access information associated with that resource **310** (by resource manager API **404**). Each resource manager **304** maintains resource information about a particular corresponding resource **310** in tabular form as a resource manager table **406** in a memory, for instance in memory **208** associated with switch controller **112**. Requestors **306** query a particular resource manager **304** using that manager's resource manager API **404** for that information. The resource manager **304** being queried uses information contained in the structure of the query from requestor **306** for retrieving specified data from resource manager table **406** and/or transacting with resource **310** (see page 18, line 9 to page 19, line 14). In sharp contrast with the prior art, resource manager APIs **404** are generic and not specialized APIs, specific to a unique resource. Requestor **306**

does not directly communicate with any of resources **310A - 310n**, but instead interfaces with resource managers **304A - 304n** contained in resource management **114**. Thus, requestors **306** need only understand the standardized generic APIs common to all to resource managers **304A - 304n** in order to access all resources' data managed by resource management routine **114**.

In accordance with one exemplary embodiment of the present invention, each unique



**FIG. 4**

resource information table **406** corresponds to a particular resource **310** and is stored in the switch controller's memory, for instance memory **118**. Similarly, every resource management API **404** corresponds to particular resource **310** and therefore, provides a unique table of information **406** for particular resource **310** is also stored in the switch controller's memory.

## VI. ISSUES (37 C.F.R. 1.192(c)(6))

The issues on appeal are whether:

(1) Claim 5 is unpatentable by virtue of being indefinite under the meaning of 35 U.S.C. § 112 second paragraph - **The issues concerning claim 5 are now moot in view of the Examiner's Answer.**

(2) Claims 1 - 8, 10, 11, 15, 16, 18 and 19 are unpatentable over Sofman (U.S. Patent No. 5,937,042) under 35 U.S.C. § 102(e). **The issues concerning claim 5 are now moot in view of the Examiner's Answer.**

(3) Claim 9 is unpatentable over Sofman (U.S. Patent No. 5,937,042) in view of Taylor, et al. (U.S. Patent No. 5,912,961) under 35 U.S.C. § 103(a).

(4) Claims 12 - 14 and 17 are unpatentable over Sofman (U.S. Patent No. 5,937,042) in view of Gottlieb (U.S. Patent No. 5,920,621) under 35 U.S.C. § 103(a).

(5) Claim 20 is unpatentable over Sofman (U.S. Patent No. 5,937,042) in view of Reto et al. (U.S. Patent No. 5,825,857) under 35 U.S.C. § 103(a).

## VII. GROUPING OF CLAIMS (37 C.F.R. 1.192(c)(7))

The claims do not stand or fall together as a single group. Instead, the claims stand and fall in seven groups as follows:

Group A -- claims 1, 4 and 19;

Group B -- claims 2, 6 and 7;

Group C -- claims 3, 8, 10, 11, 15, 16 and 18;

Group D -- claim 5 - **The issues are now moot in view of the Examiner's Answer;**

Group E -- claim 9;

Group F -- claims 12 - 14 and 17; and

Group G -- claim 20.

With regard to the Examiner's Answer, the Examiner has allowed claim 5 and as such that group has been deleted from the claim grouping in the reply brief.

In addition, the Examiner has asserted that the claims do not stand or fall together because Appellant's brief does not include a statement that this grouping of claims do not stand or fall together and "reasons in support thereof."

Appellant's representative disagrees. The Appeal Brief states that the claims do not stand or fall together, but clearly groups the 20 claims into Groups A - G and identifies the claims within each group.

As to an explanation as to why the claims are believed to be separately patentable, the argument section of the brief, Section VII of the Appeal Brief, clearly provides detailed explanations as to why the claims are believed to be separately patentable. Appellant discusses each of Groups A - G, claim limitations common to the particular claim Groups and reasons why the Groups are believed to be separately patentable. For most Groups, the brief provides multiple reasons why the Groups are separately patentable. Additionally, each Group's explanation is presented under a separate heading that identifies the claims in the particular Group.

Moreover, if the Examiner believes that the Appeal Brief did not contain the prerequisite explanations for Grouping, the MPEP requires that the Examiner set a period for curing the defect under 37 CFR 1.192(d) using form paragraph 12.78 (MPEP 1206). Nowhere does Appellant find

support for the Examiner lumping all claims into a single group for Appeal and for not providing an explanation as to why the claims are believed to be separately patentable.

**VIII.A. ARGUMENTS—REJECTIONS UNDER 35 U.S.C. 112, (37 C.F.R. 1.192(c)(8)(i))**

**I. 35 U.S.C. § 112, Second Paragraph**

With regard to the Examiner's Answer, the Examiner has withdrawn the rejections of claim 5 is under 35 U.S.C. § 112 second paragraph. The issues concerning claim 5 are now moot.

**VIII.B.        ARGUMENTS—REJECTIONS UNDER 35 U.S.C. 103, (37 C.F.R. 1.192(c)(8)(i))**

**I.        The rejection of claims 1, 4 and 19 under U.S.C. § 102 is incorrect.**

The Examiner has rejected claim nos. **1-8, 10-11, 15-16 and 18-19** under 35 U.S.C. § 102(e) as being anticipated by Sofman (U.S. Patent No. 5,937,042).

**With Regard to the Examiner's new discussion and description of the alleged prior art**

**Examiner's Answer:** The Examiner asserts that "claim **1** seems more indicative of Appellant's **FIG. 1**." The Examiner also asserts that the data processing system (fig. 3 of Sofman) could include a switch. Finally, the Examiner makes the blanket assertion that "the data processing system [of Sofman] thus reads on a computer system which the examiner considers to contain all the elements including that of (figs. 2 and 3)," and goes on to describe Appellant's **FIG. 3**.

As a threshold requirement, the Examiner is required to examine the claims, specifically at least claim **1** of Group A. Regardless of how the Examiner's characterization of the present invention, clearly Appellant's **FIG. 4** DEPICTS THE ELEMENTS OF THE RESOURCE MANAGEMENT MEANS RECITED IN CLAIM **1** and not **FIGs. 1 - 3**. **FIGs. 1, 2 and 3** each lack feature limitations required by claims **1, 4 and 19**. It is clear from the Examiner's new arguments that the limitations of, at least, claim **1** are not fully understood or appreciated by the Examiner. The Examiner is invited to reread claim **1**, the SUMMARY OF THE INVENTION in the Appeal Brief and the discussion below.

At a minimum, claim **1** requires a telecommunications network that includes both a processor (**206**) **AND** a resource management means (**114**). The claim further recites that the resource management means (**114**) comprises one or more resource managers (**304A - 304n**) for enabling the processor (**206**) to provide the "standardized" management of multiple resources (**310A - 310n**) including internal operational resources, external components, and applications processing data). Apparently, this distinction has been overlooked by the Examiner. Standardized management of multiple resources (**310A - 310n**) allows any requestor that interacts with any resource manager of the resource management means (**114**) to avail itself of the standardized management of multiple resources (**310A - 310n**). Thus, a requestor need not understand how to communicate, interact or manage any particular resource, but instead merely

understand the "standardized" management of multiple resources that is common to all resource managers of the resource management means (114).

Claim 1 further requires that each resource manager (*i.e.*, resource manager 304A - resource manager 304n) comprises one or more resource manager application program interface (API) (404A - 404n) AND one or more data storing means (406A - 406n). These logical elements are not shown in any of FIGs. 1 - 3. Moreover, the Examiner has simply not pointed to any teaching or suggestion in Sofman that describes a resource management means comprising resource managers which in turn comprises at least a resource manager API AND a data storing means. This fact is clear from the Examiner's insistence that Appellant's FIGs. 1 - 3 depict the invention as required by claim 1. These figures simply do not show the resource management interface concept of the resource management means as including resource managers that further include both management APIs and a data storage means. The Examiner's discussion of Sofman also fails to adequately point out these features as required by claim 1.

Furthermore, each resource manager API (404A - 404n) is used for managing at least one of the multiple resources (310A - 310n). Each of resource manager APIs (404A - 404n) are said to manage at least one of resources (310A - 310n) by providing a particular resource with the specialized interface that is understood by that particular resource. Moreover, because each resource manager API (404A - 404n) is a part of one of resource managers 304A - 304n, which are a part of a resource management means, each resource manager API (404A - 404n) enables "the processor (206) to provide the "standardized" management of multiple resources (310A - 310n)." Therefore, all of resource manager APIs (404A - 404n) include an interface that is standard to the resource management means (114) for interacting with, such as by requestor (306). Thus, requestor (306) need only to understand the standard interface to update data from any resource associated with resource management means (114). Without this feature, a requestor could not manage the multiple resources without understanding all of the resource APIs used by the individual resources to be managed by the requestor. Apparently, Sofman's data granulator 104 accesses data from the individual resources 122 - 130 using the resources' interfaces and not through a standard management resource API. More importantly, nowhere does Sofman describe any mechanism for a requestor to initiate a request to any resource via data granulator 104. Apparently, data granulator 104 merely accesses data at the resources and stores

it in database **106**. Thus, a requestor could not manipulate the data in database **106** to reflect the current resource state (*i.e.*, update data in database **106**), but merely pull existing data from it. This could not be fairly characterized as any type of resource management.

Additionally, resource manager APIs (**404A - 404n**) are used for accessing the resource manager's data storing means (**406A - 406n**) directly. By using the resource manager APIs (**404A - 404n**), a requestor **306** need only understand the standardized generic APIs common to all to resource managers **304A - 304n** in order to access all resources' data managed by resource management routine **114**. Therefore, claim **1** requires that each resource API support trifurcated roles: 1) the resource manager API should be understood by the particular resource being queried by the resource manager (thus adopting the resource's own API); 2) the resource manager API should be part of the standardized management of multiple resources (thus a requestor need only understand how to interact in accordance with the standardized management interface or generic APIs of the resource management means); and 3) the resource manager API should be used for manipulating data in the data storage means to reflect the current state of the resource.

When taken as a whole, claim **1** requires resource management features that are not taught or suggested by Sofman, in particular "manipulating data in the data storage means to reflect the current state of the resource." Sofman's data granulator **104** accesses network data **102** from network resources **122 - 130**, which could reflect the current resource state at the time the resource was assessed. That data is then deposited in database **106**. However, when retrieved by rehoming optimizer **108**, the data no longer reflects the current resource state but merely the last-in resource data. Moreover, rehoming optimizer **108** does not manipulate the data to reflect the current resource state, but merely retrieves the data. Because Sofman does not truly manage network resources **122 - 130**, a requestor, such as rehoming optimizer **108**, must accept whatever data is present in database **106**. On the other hand, claim **1** requires that the resource manager have one or more resource manager APIs that manage resources using standardized management. The resource manager APIs may be used for manipulating data in the data storage means to reflect the current state of the resource. Thus, a requestor can insure that the data in the data storage means reflects the true current state of a resource by invoking an update command using the resource manager API, for any resource managed by a resource

manager in the resource management means. Thus, rather than the data storage means merely being filled with last-in resource data, the data in the data storage means is manipulated to reflect the current resource state by the requestor.

The Examiner goes on to assert:

The examiner would like to point out that the data processing system thus reads on a computer system which the examiner considers to contain all the elements including that of (figs. 2 and 3).

Furthermore, the examiner would like to point the appellant to fig. 3 which teaches a resources management (see memory and column 7 lines 21-30, column 5 lines 61-67) which would contain programming applications. Now, the resource management means (programs) with inherently multiple instructions or codes provides management of a plurality of the data processing system's resources including hard drives, all the other components (hardware components) and also, raw data gathered during the optimization process which can be formatted into tables as shown in (figs) of the prior art which reads on a resource management enabling a processor to provide standardized management of multiple resources including internal operations resources, external components and application processing data.

Apparently, the Examiner intends to distill the Appellant's invention down to inherently understood elements within Sofman. Even if there were a program in Sofman's computer system that somehow equated to the resource management means, although not expressly mentioned, the Examiner's logic is faulty. Claim 1 requires that the separate resource managers manage the system resources and NOT the resource management means. Moreover, nothing in Sofman suggests modifying system 200 in this manner toward the presently-claimed invention.

The Examiner continues by asserting "if the data processing unit is considered as a single computer system, then the components including the data granulator can be considered a resource manager which according to (column 18 lines 25-31) can use a standard query language (SQL), a computer language. First, Appellant's representative reminds the Examiner that the present claims have been rejected under 35 USC §102 and not under §103. Therefore, each claim limitation must be present in the alleged prior art reference and not constructed from individual elements in a manner never intended by the reference. Nevertheless, if the Examiner's assertions are correct, how is SQL used by data granulator 104 to "manage said internal operational resources, said external components, and said applications processing data" as

required by claim 1? Apparently, the Examiner is referring to how the data is organized in database 106 allowing rehomeing optimizer 108 to issue SQL queries for data retrieval. While it may be true that a database may be organized into SQL database tables, there is no teaching in Sofman that SQL is used to manage the separate resources as insisted by the Examiner. Nothing in Sofman suggests that SQL queries are understood by any of the network resources 122 - 130. Moreover, if the network resources understood SQL queries, then the role of data granulator 104 would be redundant. In that case, the rehomeing optimizer 108 could access the data directly from network resources 122 - 130 using SQL queries and circumvent data granulator 104. This is simply not taught or suggested by Sofman.

The Examiner continues by stating:

... also, C+ plurality of possibly interfacing a data processing system "computer" with a plurality of other systems "adapters, peripheral elements and so forth" which would be able to communicate with teach [sic] other. The prior art (Sofman) teaches displaying current state of network resources (RCGs) in conjunction with traffic data which can be stored in one of the storage means of the data processing system.

Appellant's representative again reminds the Examiner that the present claims have been rejected under 35 USC §102 and not under §103. Sofman does not teach these limitations, but instead are mere suppositions made by the Examiner that must be responded to. From the outset, it should be understood that merely because two devices' software systems are implemented in the same programming language does not mean that they can communicate with each other. Moreover, the present claims require management of external resources as well as internal resources making it even less likely that the disparate systems could communicate at the low level of a programming language. The fact that the system components support C+, or any other programming language, does not suggest any standardized management of multiple resources. If the Examiner's logic were adopted by artisans in the art, then all boundaries between systems could be traversed by low level programming languages. This is simply not so.

Moreover, the fact that C+ based "adapters, peripheral elements and so forth" communicate with one another merely restates the problems encountered in the admitted prior art which is that each resource requires a separate API, interface or some other adapter that it understands. These interfaces are often resource-specific, *i.e.* supplied by the resource.

Therefore, any requestor that intends to interact with a resource would need that resource's adapter and understand it. Thus, a requestor might find it necessary to acquire dozens or maybe hundreds of resource-specific adapters or interfaces to get the current resource states for the separate resources. This is NOT suggestive of "resource management means for enabling said processor to provide standardized management of multiple resources including internal operational resources, external components" as required by the claims.

Moreover, by the Examiner's own admission, Sofman does not teach the present invention as required by claim 1. Recall that claim 1 requires a "... resource manager APIs that manage" a plurality of resources AND for "manipulating the data (in the storage means) to reflect the current resource state." Assuming, *arguendo*, that all of the inherent features suggested by the Examiner are in Sofman, and even considering the structure of system 200 in the manner proposed by the Examiner, nowhere does Sofman suggest an adapter, interface or any other means for both accessing data from the separate resources AND depositing the resource data in database 106. By the Examiner's own supposition, Sofman would use C+ for communications between a resource manager and a resource and then SQL to load the database. The claims require that both as facilitated by a resource manager API. Therefore, by the Examiner's own admission, Sofman cannot teach the present invention as required by claim 1.

## **II. The rejection of claims 2, 6 and 7 under U.S.C. § 102 is incorrect.**

**With Regard to the Examiner's new discussion and description of the alleged prior art Examiner's Answer:** The Examiner bases the rejection of claim 2 on the logic provided above for claim 1 that has been overcome. Additionally, the Examiner now intends that rehoming optimizer 108 and not data granulator 104 is a resource manager, so the Examiner's logic necessarily cannot follow that supposed in the discussion of claim 1.

The Examiner now asserts that: "'sending a query to a resource manager wherein the resource manager manages information corresponding to a resource, complying with a common standard for resource managers within the network' could read on a system administrator sending rehoming criterion from an interface to a rehoming optimizer for which subsequent answers solution would be provide in regard to information pertinent to the network such as resource information shown in (figs. 28-30)." This is apparently an entirely new ground for rejecting the

claims and the Examiner's conclusions are completely unsupported by the reference. Appellant's representative reminds the Examiner of the probation against the entry of new grounds of rejection in the Examiner's Answer (MPEP 1208.01). The Examiner had ample opportunity during the extended prosecution of this application to forward any contention of a manual implementation of the present claims but chose not to.

It is not clear that the rehomeing optimizer taught by Sofman could support the system suggested by the Examiner, much less teaches the specific limitations. Regardless, in addition to requiring a standardized resource manager claim 2 requires "managing data stored in memory and organized in table format using said query, including manipulating the data to reflect the current resource state," which rehomeing optimizer 108 cannot do. At best, rehomeing optimizer 108 merely sends SQL queries to database 106. Nowhere does Sofman suggest that rehomeing optimizer 108 can manage "data stored in memory and organized in table format using said query" or manipulate "the data to reflect the current resource state." At best, rehomeing optimizer 108 retrieves a data image of whatever data is available to it in database 106. Nowhere does Sofman teach that rehomeing optimizer 108 manages data in database 106. If any component manages the network data in database 106, Sofman teaches that data granulator 104 organizes the data as normalized SQL tables -NOT rehomeing optimizer 108.

Moreover, even if rehomeing optimizer 108 could manage "data stored in memory," rehomeing optimizer 108 surely cannot manipulate "the data to reflect the current resource state." Here, the Examiner confuses "displaying" data on a display screen with "manipulating the data to reflect the current resource state." If, as is hoped by Sofman, the data in database 106 reflects the current resource state of resources 122 - 130, a copy of the data is accessed and used in the rehomeing optimization process. If not, whatever data is in database 106 is accessed and used in the rehomeing optimization process. In any case, rehomeing optimizer 108 is stuck with whatever data is available in database 106. Nothing in Sofman suggests that in the event that rehomeing optimizer 108 detects stale data in database 106 it manipulates "the data to reflect the current resource state." In order to manipulate "the data to reflect the current resource state," a query from rehomeing optimizer 108 must be used to query the resource itself, such as by using the resource manager API of the present invention. Sofman neither teaches nor suggests these limitations. What is more, if any query from rehomeing optimizer 108 could manipulate "the data to reflect the current resource state," then data granulator 104 would be superfluous because

rehoming optimizer **108** itself could manipulate the data to reflect the current resource state. Data granulator **104** would be unnecessary for accessing resource data. So much for the manual interpretation.

With regard to the Examiner's second supposition, the Examiner supposes:

Another interpretation could be a data granulator (see fig. 2), an element of the data processing system, a first resource manager, "accesses" one of a second resource managers: data repositories (see column 4 lines 42-54). According to (column 18), a common standard namely; SQL (standard query language) can be used a common standard, in creating a table detailing current resource state (see column 2 lines 1-16) and column 18 lines 25-31) based on gathered switch data. The appellant limits a resource manager to one component but the examiner begs to differ because a system could comprises of a plurality of resource managers (processors) functioning together to make a system operable. An example could be a first resource manager (processor) communicating with another manager (manager) as taught by Sofman. Furthermore, see the explanation as set forth in the rejection of claims 6 and 7.

The meaning of this interpretation is not completely understood by Appellant's representative; however, as best understood, nothing in Sofman suggests this interpretation. Even if this supposition could be supported, Appellant's representative again reminds the Examiner that the present claims have been rejected under 35 USC §102 and not under §103. Moreover, this interpretation appears also to express new ground for rejecting the claims. Appellant's representative reminds the Examiner of the probation against the entry of new grounds of rejection in the Examiner's Answer (MPEP 1208.01). Here again the Examiner had ample opportunity during the extended prosecution of this application to forward any contention of multiple resource managers interacting with one another to form a basis for anticipating the present claims. Here again the Examiner did not develop this argument during prosecution.

As to the specific allegation, the Examiner apparently contends that data granulator **104** accesses repositories associated with resources **122 - 130**. However, according to the Examiner, for this to work resources **122 - 130** are now resource managers themselves instead of resources as indicated by Sofman. Regardless, using the Examiner's protocol data granulator **104** accesses repositories associated with resources **122 - 130** using SQL queries because, apparently, these repositories consist of normalized SQL database tables. Wow! The Examiner has met the claim limitations by merely assuming that resources **122 - 130** are not resources but resource managers, providing SQL structure to the alleged resource manager's databases and supposing that data

granulator **104** makes SQL queries to the alleged resource managers. The problem with the Examiner's logic is that the resource managers that met the claim limitations are those alleged from resources **122 - 130** and Sofman does not discuss those elements in detail. Clearly, here the Examiner is improperly reading the claim limitation into the teaching of Sofman. Here again, if the repositories associated with resources **122 - 130** were structure SQL databases, then rehoming optimizer **108** could merely access their data directly rather than relying on data granulator **104** to fill database **106**. At best, this supposition by the Examiner is not supported by Sofman.

**III. The rejection of claims 3, 8, 10, 11, 15, 16 and 18 under U.S.C. § 102 is incorrect.**

**With Regard to the Examiner's new discussion and description of the alleged prior art Examiner's Answer:** the Examiner apparently relies on Sofman's mention of a semaphore as basis to anticipate the claim. The Examiner states:

The appellant argues that the prior art of record to teach the claimed limitations of claim 3. The examiner disagrees because the plurality of application interface means are nothing more but programmable codes or part of a program such as semaphore (see claim 9) taught by Sofman (column 19 line 10), set up or print operational measurement (see claim 11), a feature taught by Sofman (see figs, column 13, column 15 lines 16-28) which teaches being able to set up, delete add or print setup; and to be able to create a group entry (see claim 15), something Sofman teaches by giving the user the ability to create group entry (RCG group) through inherent software modules (see figs.). Sofman teaches a plurality of applications which makes the data processing system function. Sofman teaches at least one resource requestor (program module) interacting with other resources (see column 21). Sofman teaches that a standard language including SQL, common language code or a language such as C+ could be used in building a resource including software modules in controlling a data processing system. The switch taught by Sofman serves to provide a plurality of functionalities including call completion, traffic monitoring and so forth. Sofman in (columns 21-22) teaches a plurality of processes including matching, validation, sorting, selecting and so forth when resources are requested.

Here again, the Examiner's remarks are not well understood. Apparently the Examiner now offers another new ground for rejecting the claims by way of analogizing a semaphore to the claimed application program interface. This too is apparently an entirely new ground for rejecting the claims and here again the Examiner's conclusions are completely unsupported by the reference. Appellant's representative on again reminds the Examiner of the probation against the entry of new grounds of rejection in the Examiner's Answer (MPEP 1208.01). The Examiner

had an ample opportunity during the extended prosecution of this application to forward any contention of a semaphore being analogous to the claimed application program interface but did not. More importantly, this analogy is not supported by the present specification, the claim limitations, the alleged prior art or any reasonable definition of the functionality of a semaphore. Here the Examiner's analogy could have been easily dispensed with during prosecution if the assertion had been properly raised by the Examiner.

During prosecution the Examiner merely pointed to passages of the alleged prior art while making few, if any, specific allegations supporting the rejection of specific claim limitation. Appellant's representative was left to formulate every conceivable argument in favor of the Examiner's position, make the substantive arguments for the Examiner and then, demonstrate that any reasonable argument that the Examiner could have made would not have met the legal burden. Now, once that prosecution has ended, the Examiner makes a completely illogical assertion, that is unsupported by the alleged prior art, the claims or even the generally accepted definitions within the art and expects the rejection to be affirmed by the Board. This tact is completely improper.

However, claim 3 requires that a plurality of application program interface means, each providing an interface between one or more resource requestors and data organized in a table corresponding to one of a plurality of resources. Each application program interface means comprising both a sending means for sending a query and a managing means for managing data stored in said memory using said query. Therefore, unless the Examiner can point to "programmable codes or part of a program such as semaphore" that provides an interface between a resource requestors and data organized in a table corresponding to a resources, the Examiner has not met the burden. The Examiner points to claim 9 for support; however, it should be clear that claim 9 (see also page 27, line 18 to page 28, lines 5 in the specification) treats semaphores as resources. Semaphores act as the gatekeepers of the memory by preventing simultaneous access to a memory segment by more than one process. Therefore, rather than a semaphore being illustrative of an application program interface, as proffered by the Examiner, a semaphore is a resource to be managed by a semaphore resource manager. A semaphore resource manager has a plurality of semaphore APIs for enabling a processor to provide an interface between a resource requestor and tabular data in a memory corresponding to the semaphore resource. Each semaphore API includes a sending means for sending a query and a

managing means for managing data stored in the memory the query. Nowhere does Sofman even suggest such a semaphore. The Examiner's logic does not support this realization.

In addition, the Examiner alludes to functionality related to "being able to set up, delete add or print setup; and to be able to create a group entry." However, the Examiner never makes a connection in Sofman on just which, if any, API performs these functions and even if a API exists, or that ant API meets the requirements of at least claim 3. As best as can be understood, the Examiner's discussion is simply not relevant to claim 3.

**IV. The rejection of claim 5 under U.S.C. § 102 is incorrect.**

With regard to the Examiner's Answer, the Examiner has withdrawn the rejections of claim 5 is under 35 U.S.C. § 112 second paragraph and 35 U.S.C. § 102. The issues concerning claim 5 are now moot.

**VIII.B. ARGUMENTS—REJECTIONS UNDER 35 U.S.C. 103, (37 C.F.R. 1.192(c)(8)(i))**

**V. The rejection of claim 9 under U.S.C. § 103 is incorrect.**

**With Regard to the Examiner's new discussion and description of the alleged prior art Examiner's Answer:** The Examiner has rejected claim 9 under 35 U.S.C. § 103(a) as being unpatentable over Sofman (U.S. Patent No. 5,937,042) in view of Taylor, et al. (U.S. Patent No. 5,912,961).

The Examiner states:

The appellant argues that the combination is improper and lacks motivation. The examiner disagrees, a second look, at the primary reference (Sofman) in fact teaches semaphores (see column 19 line 10) which according to Sofman enforces the correct order of processing.

Claim 9 requires one of any management resource API listed related to a semaphore resource. The Examiner apparently has confused the resource with its API. The Examiner's remarks are not relevant to the claim.

**VI. The rejection of claims 12 - 14 and 17 under U.S.C. § 103 is incorrect.**

The Examiner has rejected claim nos. **12 - 14** and **17** under 35 U.S.C. § 103(a) as being unpatentable over Sofman (U.S. Patent No. 5,937,042) in view of Gottlieb (U.S. Patent No. 5,920,621).

The appellant argues that rejection of claims 12-14 and 17 lacks motivation to combine both references. The examiner disagrees because even though, Sofman is directed to traffic management system, the system is also directed to a telephone communication system used in completing for telephone calls. For instance, in (column 8 line 47), Sofman mentions a "calling card feature" thus it could be said incorporating other known telephone features into that of the switch if view from a larger context would be obvious ie: a switch capable of providing telecommunications services in conjunction with the fact that it can monitor call traffic data.

Claims **12 - 14** and **17** claim different types of resource manager APIs, as do claims **10, 11, 15** and **16** that stand rejected under 35 U.S.C. § 102. Nowhere can Appellant's representative find any mention of one of the resource manager APIs, or even, for the most part, the resource type itself. While Gottlieb does suggest similar resources, Gottlieb does not teach or suggest resource manager APIs of the type claimed.

**VII. The rejection of claims 20 under U.S.C. § 103 is incorrect.**

The Examiner has rejected claim **20** under 35 U.S.C. § 103(a) as being unpatentable over Sofman (U.S. Patent No. 5,937,042) in view of Reto et al. (U.S. Patent No. 5,825,857).

Heading VII in the Appeal Brief contained an obvious typographical error. That error was not transposed in the body of the section and the argument submitted there remain accurate. The Examiner apparently has no comment on the remarks contained within that section. The above heading has been corrected.

### VIII. Conclusion

In view of the additional comments above, it is respectfully urged that the Examiner's rejection of the claims should not be sustained.

Respectfully submitted,



Rudolph J. Buchel Jr.\*

Reg. No. 43,448

***Jones, Day, Reavis & Pogue***

P.O. Box 660623

Dallas TX 75266

214 969-2990

214 969-5100 Fax

Date: February 12, 2002

Attorney for Appellant

*\*Licensed in Florida*

*Not Licensed in Texas*